**SSN COLLEGE OF ENGINEERING (Autonomous)**

**Affiliated to Anna University**

**DEPARTMENT OF CSE**

**UCS 1312 Data Structures Lab Laboratory**

**EX3:** **Implementation of doubly linked list ADT**

**=====================================================================================REGISTRATION NO: 185001112**

**NAME: PRATHYUSH S**

**CLASS: CSE-B (SEMESTER-3)**

**=====================================================================================Aim:**

To Create a doubly linked list to store set of student names Perform the following operations using a menu driven program

1. Insert student name in the front of the list

2. Insert student name at the end of the list

3. Insert a record after a given name in the list

4. Search a given student in the list

5. Delete a given student

6. Display all student names

7. Display the students in alphabetical order

**Note:**

To maintain 3 files.

1. Structure and function definitions – **function.h**

2. Function prototypes – **prototype.h**

3. Application – **main.c**

**Source Code:**

**1) function.h**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct mynode

{

char name[30];

struct mynode \*prev,\*next;

}node;

node\* create()

{

node \*head,\*tail;

head=(node\*)malloc(sizeof(node));

tail=(node\*)malloc(sizeof(node));

head->prev=NULL;

head->next=tail;

tail->prev=head;

tail->next=NULL;

return head;

}

void insertbeg(node \*head)

{

node \*temp;

char nm[30];

temp=(node\*)malloc(sizeof(node));

printf("\nEnter name to be inserted first: ");

scanf("%s",nm);

strcpy(temp->name,nm);

temp->next=head->next;

temp->prev=head;

head->next->prev=temp;

head->next=temp;

printf("\nName inserted successfully at the beginning\n");

}

void insertend(node \*head)

{

node \*temp,\*tmp,\*ptr;

char nm[30];

temp=(node\*)malloc(sizeof(node));

tmp=(node\*)malloc(sizeof(node));

ptr=(node\*)malloc(sizeof(node));

ptr=head;

printf("\nEnter name to be inserted last: ");

scanf("%s",nm);

while(ptr->next!=NULL)

ptr=ptr->next;

tmp=ptr;

strcpy(temp->name,nm);

temp->next=tmp;

temp->prev=tmp->prev;

tmp->prev->next=temp;

tmp->prev=temp;

printf("\nName inserted successfully at the end\n");

}

int findpos(node \*head,char nm[])

{

int i=1;

node \*temp;

temp=(node\*)malloc(sizeof(node));

temp=head->next;

while(temp!=NULL && (strcmp(nm,temp->name)!=0))

{

temp=temp->next;

i++;

}

if(temp==NULL)

{

printf("\nStudent name is not found\n");

return -1;

}

--i;

return i;

}

void insertpos(node \*head,char bef[])

{

int pos=0,i;

node \*temp,\*ptr;

char nm[30];

temp=(node\*)malloc(sizeof(node));

temp=head;

ptr=(node\*)malloc(sizeof(node));

pos=findpos(head,bef);

if(pos!=-1)

{

printf("\nEnter the student name: ");

scanf("%s",nm);

strcpy(ptr->name,nm);

pos+=2;

for(i=1;i<pos;i++)

temp=temp->next;

ptr->next=temp->next;

ptr->prev=temp;

temp->next->prev=ptr;

temp->next=ptr;

printf("\nName inserted successfully\n");

}

}

void search(node \*head,char nm[])

{

int c;

c=findpos(head,nm);

if(c!=-1)

printf("\nStudent name is in %d position\n",(++c));

}

void dellist(node \*head)

{

int p=0,i;

char nm[30];

printf("\nEnter the student name: ");

scanf("%s",nm);

p=findpos(head,nm);

if(p!=-1)

{

node \*temp,\*ptr;;

temp=(node\*)malloc(sizeof(node));

temp=head;

ptr=(node\*)malloc(sizeof(node));

for(i=0;i<p;i++)

temp=temp->next;

ptr=temp->next;

strcpy(ptr->name,nm);

temp->next=temp->next->next;

temp->next->prev=temp;

free(ptr);

printf("\nName Deleted successfully\n");

}

}

void display(node \*head)

{

node \*temp;

temp=(node \*)malloc(sizeof(node));

temp=head->next;

printf("\nStudent records\n");

while(temp->next!=NULL)

{

printf("\n%s",temp->name);

temp=temp->next;

}

printf("\n");

}

int overall(node \*head)

{

int count=0;

node \*temp;

temp=(node\*)malloc(sizeof(node));

for(temp=head->next;temp->next!=NULL;temp=temp->next)

count++;

return count;

}

void sort(node \*head)

{

int a, i;

node \*tmp,\*ptr;

tmp=(node\*)malloc(sizeof(node));

ptr=(node\*)malloc(sizeof(node));

char str[100];

a=overall(head);

for(i=0;i<a;i++)

{

for(tmp=head->next;tmp->next->next!=NULL;tmp=tmp->next)

{

for(ptr=tmp->next;ptr->next!=NULL;ptr=ptr->next)

{

if (strcmp(tmp->name,ptr->name)>0)

{

strcpy(str,tmp->name);

strcpy(tmp->name,ptr->name);

strcpy(ptr->name,str);

}

}

}

}

printf("\nStudents names are arranged in alphabetical order\n");

display(head);

}

**2) prototype.h**

#include "student\_dll.h"

node\* create();

void insertbeg(node \*head);

void insertend(node \*head);

int findpos(node \*head,char nm[]);

void insertpos(node \*head,char bef[]);

void search(node \*head,char nm[]);

void dellist(node \*head);

void display(node \*head);

int overall(node \*head);

void sort(node \*head);

**3) main.c**

#include <stdio.h>

#include "prototype\_dll.h"

int main()

{

node \*head;

int ch;

char arr[30],c;

head=create();

do

{

printf("\n\n1.Insert Student Name in the front\n2.Insert Student Name at the end\n3.Insert a Student Name after a given name\n4.Search a given Student Name\n5.Delete a given student Name\n6.Display all Student Names\n7.Display the Student Names in alphabetical order.\n8.Exit\n");

printf("\nEnter choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:

insertbeg(head);

break;

case 2:

insertend(head);

break;

case 3:

printf("\nEnter name to be inserted after whom?: ");

scanf("%s",arr);

insertpos(head,arr);

break;

case 4:

printf("\nEnter the name of the person to be searched:");

scanf("%s",arr);

search(head,arr);

break;

case 5:

dellist(head);

break;

case 6:

display(head);

break;

case 7:

sort(head);

break;

case 8: printf("Thank you!!!");

break;

default:

printf("\nInvalid choice\n");

}

}while(ch!=8);

return 0;

}

**Output:**

gml36:Desktop cseb64$ gcc doublelinkedlist.c -o dll

gml36:Desktop cseb64$ ./dll

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:1

Enter name to be inserted first: Prathyush

Name inserted successfully at the beginning

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:2

Enter name to be inserted last: Kumar

Name inserted successfully at the end

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:3

Enter name to be inserted after whom?: Prathyush

Enter the student name: Praveen

Name inserted successfully

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:3

Enter name to be inserted after whom?: Prathyush

Enter the student name: Bharath

Name inserted successfully

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:4

Enter the name of the person to be searched:Bharath

Student name is in 2 position

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:5

Enter the student name: Bharath

Name Deleted successfully

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:6

Student records

Prathyush

Praveen

Kumar

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:7

Students names are arranged in alphabetical order

Student records

Kumar

Prathyush

Praveen

1.Insert Student Name in the front

2.Insert Student Name at the end

3.Insert a Student Name after a given name

4.Search a given Student Name

5.Delete a given student Name

6.Display all Student Names

7.Display the Student Names in alphabetical order.

8.Exit

Enter choice:8

Thank you!!!